Exhibit III

```perl
#!/usr/local/bin/perl
# COPYRIGHT   (c) 1998 Bell Communications Research Inc.,
#    All Rights Reserved.
#
# PROPRIETARY - BELLCORE AND AUTHORIZED CLIENTS ONLY.
#
# This document contains proprietary information that shall
# be distributed or routed only within Bell Communications
# Research (Bellcore), and its authorized clients, except
# with written permission of Bellcore.
#
# $Id: classify.pl,v 1.1 1999/05/05 13:20:39 rmartija Exp $
#

undef;
require 'getopts.pl';

$USAGE = "Usage: " . $0 . " [-D] -p path -d file -t type -h file\n\n" .
        "Options:\n" .
        "    -D          debug mode\n" .
        "    -p path     output directory\n" .
        "    -d file     name of domains file\n" .
        "    -t type     domain type (1=US only, 2=Non-US only, 3=Global)\n" .
        "    -h file     name of hosts file\n\n" .
        "Example:\n" .
        "    $0 -p ../data/local -d alldomains.lcl -t 1 -h dat.txt\n" .
        "    $0 -D -p ../data/national -d alldomains.nat -t 2 -h dat.txt\n" .
        "    $0 -p ../data/non-us -d alldomains.nus -t 3 -h dat.txt\n\n" ;


#-------------------------------------------------------------------
#-------------------------------------------------------------------
sub prompt {
    my( $msg, $choices, $default, $nocase ) = @_;
    my( $reply );

    print STDERR $msg;

    print STDOUT " ($choices)" if( $choices );
    print STDOUT "? ";
    print STDOUT "[$default] " if( $default );

    $reply = <STDIN>;
    chop( $reply );

    ($reply =~ tr/a-z/A-Z/) if( $nocase );
    ($reply = $default) if( length($reply) == 0 && $default );

    return $reply;
}


#-------------------------------------------------------------------
#-------------------------------------------------------------------
sub overwrite {
    my( $file ) = @_;
```

```
        return 1 if( ! (-e $file || -d $file) );

        my( $msg ) = "$file exists. Overwrite";
        return 1 if( &prompt($msg, "Y/N", "Y", 1) eq "Y" );
        return 0;
    }


###########################################################
#################### main program #########################
###########################################################

    $x = &Getopts( 'd:h:p:t:D' );
    die "$USAGE\n" unless ($x ne '');

    die "ERROR: cannot open $opt_d\n" unless -r $opt_d;
    die "ERROR: cannot open $opt_h\n" unless -r $opt_h;
    die "ERROR: domain type not specified\n$USAGE\n" unless $opt_t;

    die "ERROR: invalid domain type ($opt_t)\n$USAGE\n"
        unless (int($opt_t) >= 1 && int($opt_t) <= 3);

    chop( $basket = `basename $opt_d` ) ;
    open( F, "< $opt_d" ) ;
    @domains = <F> ;
    close( F ) ;

    unlink "$opt_p/$basket" unless $opt_t == 3;
    foreach $domainName (@domains) {
        chop( $domainName ) ;
        $cmd = "grep \'\\.$domainName\$\' $opt_h | cut -d\":\" -f1 " ;

        if( $opt_t == 3 ) {
            $cmd .= "> $opt_p/$domainName" ;
        }
        else {
            $cmd .= ">> $opt_p/$basket" ;
        }

        `$cmd`;
    }
```